(4)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

VLSI PUBLICATIONS

COMPUTER-AIDED FABRICATION SYSTEM IMPLEMENTATION

Semiannual Technical Report for the period October 1, 1987 to March 31, 1988

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Principal Investigator:	Paul Penfield, Jr.	(617) 253-2506
Individual Investigators:	Dimitri A. Antoniadis	(617) 253-4693
	Stanley B. Gershwin	(617) 253-2149
	Emanuel M. Sachs	(617) 253-5831
	Stephen D. Senturia	(617) 253-6869
	Donald E. Troxel	(617) 253-2570

This research was sponsored by Defense Advanced Research Projects Agency (DoD), through the Office of Naval Research under Contract N00014-85-K-0213.

TABLE OF CONTENTS

Selected Publications (starting after page 9)

E. M. Sachs and G. Prueger, "Process Model Construction and Optimization Using Statistical Experimental Design," Proceedings of Manufacturing International '88, Atlanta, GA, April, 1988. Also, MIT VLSI Memo No. 88-442, March, 1988.

D. S. Boning and D. A. Antoniadis, "A Workstation Approach to IC Process and Device Design," IEEE Design and Test of Computers, vol. 5, no. 2, pp. 36-47, April, 1988.

* T. A. Lober, J. Huang, M. A. Schmidt, and S. D. Senturia, "Characterization of the Mechanism Producing Bending Moments in Polysilicon Micro-Cantilever Beams by Interferometric Deflection Measurements," to be presented at the 1988 Hilton Head Workshop on Solid-State Transducers, June, 1988.

* Abstract only. Complete version available from Microsystems Research Center, Room 39-321, MIT, Cambridge, MA 02139; telephone (617) 253-8138.

RESEARCH OVERVIEW

Prior to 1978, the art of IC design was restricted to those who understood the electrical behavior of transistors. This did not include computer architects. Then came the Mead-Conway revolution, which opened up IC design to others. How did this happen? Three things were needed: an abstract model of a transistor without any physics; a notation for specifying a layout (CIF); and a foundry to convert CIF into chips. This revolution has spawned interesting new computer architectures, totally new circuits that were not anticipated then, and extensive development of tools for simulation, analysis, and synthesis of circuits. Although this revolution was initially considered impossible by practitioners, the new design style has turned out to be useful for all except very high-volume parts. The result is that circuit and system designers and computer architects can get custom parts by mail, for rapid prototyping. (Moreover, even high-volume parts can be designed using the new tools, and different ASIC designs can be fabricated at the same time, using the same process, in a single facility.)

A similar revolution is needed for the design of IC fabrication processes. Today this art is restricted to those who understand the physics and chemistry of the process steps. This does not include computer architects or circuit designers. What is needed is a set of absuract models of process steps, a language in which to denote process flows, and a programmable foundry to convert a layout with an associated flow to a chip. This revolution will spawn interesting new processes, novel microstructures incorporating devices other than transistors, and development of tools for simulation, analysis, and synthesis of machines, processes, and devices. This goal is considered impossible by practitioners today, but we expect that it will turn out to be useful for designing all but the most advanced processes. The result will be that circuit and system designers and computer architects can get custom parts using custom processes by mail, for rapid prototyping. (Moreover, even advanced processes will be able to be designed more rapidly using the new tools, and more than one ASP, application-specific process, can be run simultaneously in a single facility.)

The purpose of the work reported on here is to design, develop, implement, and deploy information-management systems to aid in the fabrication of integrated circuits, particularly in the context of flexible manufacturing. The work includes the development of a hardware-software system, named CAFE, and the support of this program and its use in the MIT Integrated Circuits Laboratory. CAFE should support both the manufacturing of ICs and the design of processes, so part of the effort is directed toward the development of a suitable process-development environment. Other projects are concerned with equipment and mechanical-property models and scheduling. ? [illegible] statistical experimental [illegible]

(AT ([illegible] Process Formulation) Scheduling)

mathematical models. (LP)

CAF SYSTEM STRUCTURE

The primary computer used for the actual operation of the CAF system within the integrated circuit processing facility is a Sun 3/280. Terminals are interfaced to diskless terminal concentrators, which in turn provide network connections to any available computer. Another Sun 3/280 is used as a development test bed for enhancements, tests, and debugging fixes. This insulates the users of the primary CAF system from the introduction of untested software, and it also helps to provide faster response time for the present users in that they do not have to share their cpu resources with the developers of new software. The second Sun 3/280 allows the developers of new software in particular to experiment with data base application programs without corrupting the actual data base used for ongoing processing of integrated circuits. A third computer system consisting of a DEC 785 serves as a common meeting ground for the somewhat wider community of people concerned with integrated circuit design and manufacture. The 785 also provides a connection to an MIT Physical Plant computer and maintains an active alarm log for the building containing the integrated circuits fabrication facility.

In previous quarters this year we installed the two Sun 3/280s along with a commercial version of RTI INGRES and ported all of the CAF software to these computers. Using our generalized forms based user interface, FABFORM, we implemented a new version of the equipment-reservation program which enables users to sign up for multiple machines at once. We also used FABFORM to implement a new multi-level menu system for the cafe shell. We completed initial versions of our process-flow language (PFL) together with walk through and fabrication interpreters and a browser, all of which utilize FABFORM. We encoded the baseline CMOS process in our PFL.

This quarter, we made substantial progress in the development of our data model and schema. Our Gestalt system architecture provides a uniform query interface to data residing in multiple autonomous, heterogeneous data bases. Our functional data model provides support of extended data types including various temporal types, as well as inexact, interval, and null values. We expanded our schema to represent more aspects of plant and process management: fabrication facilities and equipment, users, equipment reservations, lots, lot tracking, wafers, process flow descriptions, wip tracking, and lab activity information.

The Gestalt data base interface routines were rewritten and expanded, and a new release (complete with an updated documentation paper) made.

We have started a project to provide a schema display to let laboratory managers and users give more meaningful feedback about our schema.

We have completed a "data base walker" (DBW) program which enables application programmers (and others) to find their way around the existing data base. It displays an existing entity and allows the user to explore related entities. For example, one can display a facility and see that it has a list of machines. From there one can display a particular machine and see its attributes, etc.

We have written and released a new change status data entry program which is based on FABFORM.

We have developed, but not yet released, a generalized graphing program. This too, is based on FABFORM and allows the user to conveniently specify captions, axis labels, etc. This information, along with a data file, is then provided to the locally developed "giraphe" programs which then produce the output graphs in a form suitable for terminal display or laser printer.

We have written and are now testing a generalized equipment uptime report generator. This is designed to provide the data to the graphing programs described above. It can produce graphs of uptime for a selected machine for a selected period of time or, alternatively, a summary of the relevant log entries relating to machine status changes.

A new program which interfaces with the Nanospec has been developed. It operates the Nanospec via the computer, initiates film-thickness measurements, and places the results in the appropriate field automatically, thereby reducing the operator interaction required.

We have initiated a project to interface the Gyrex mask maker directly to the computer. Presently users write data to magnetic tape and carry these tapes to the Gyrex.

We have initiated and made substantial progress on the development of a "hands off terminal." We chose a commercially available TI speech recognition module which plugs into an IBM PC/XT. Software has been developed to interface the TI PC software to control a FABFORM interface. As this speech recognition module is speaker dependent, the software automatically loads the data base appropriate to the login name. Several of us have "trained" the recognition software and the results are quite interesting. It remains to be seen if this recognition scheme is powerful enough to actually be useful in the fabrication laboratory, at least with this hardware.

We have continued to progress on the development of a process-flow language (PFL). The creation of a PFL and associated interpreters is the key to our approach for generating actual fabrication instructions and for collecting the data resulting from actual fabrication steps. The interpreters provide the actual meaning of the process flows expressed in the flow language.

Our previous PFL development was based on only the machine setting view, in order to get something working as soon as possible. We now have a version of our PFL which is based on the two-stage process-step model which relates the goal of a change in wafer state first to the physical treatment parameters and finally to the actual machine settings used to process the wafers. We have recoded the CMOS baseline process in this new version and, in addition, have encoded a furnace monitor process which process is routinely used every week.

Besides a fabrication interpreter for this new version of the PFL, we have the rudiments of a simulation interpreter.

We have come to realize that we must provide for operation of partial flows. At least one impediment to the use of our PFL is that users change their minds about the process specification as they do the actual fabrication. By concatenating the processing history of fabrication with a number of partial flows we at least will have a trace which accurately reflects what happened.

We have made substantial progress on an expert PFL editor. This editor uses FABFORM as the user interface. Ideally one starts with an existing process flow, encoded in our lisp-like PFL syntax, which is somewhat similar to the desired process flow. The editor then displays this existing process flow with a forms based presentation and allows the user to modify the flow. The editor then produces the new flow encoded in PFL without the user even being aware of the lisp nature of the PFL. The editor supports the three views required by the two-stage generic process model and, in addition, allows any number of hierarchical levels of process-flow definition.

MODULAR PROCESS

In the last six months, we have experimented with a "Profile Interchange Format" (or PIF) for the exchange of geometry and attribute information about IC structures. Prior work was based on an ASCII or "intersite" representation of the PIF, as recently outlined by Steve Duvall of Intel. Our previous experimentation with this ASCII PIF highlighted the unsuitability of the format for direct use by simulation tools.

We have recently been working, then, toward an "intertool" version of the PIF. An interface to Gestalt forms the basis of the "intertool" or "database" form of the PIF. Utilities based on the PIF database include a Suprem-III to PIF database translator (sup2db) for stuffing the results of process simulation into the database, and a database to Suprem-III program (db2sup) for generation of the wafer structure as demanded by Suprem-III. In addition, we have developed a limited intersite PIF parser which translates the ASCII format into the database PIF (pif2db).

We are continuing development of the PIF database interface for direct use by a wide variety of process development tools, as well as pursuing implementation of tools based on this PIF database. The tools that must have access to wafer information, and thus to the PIF database, include not only process simulators such as Suprem-III, but also device simulators (such as MINIMOS), process flow language interpreters, grid manipulation programs, and analysis and plotting utilities.

We are continuing development of a CAD environment for the design of fabrication processes. This environment must be based on solid representations of the two "objects" being designed: structures to be fabricated, and fabrication processes. We have been working to lay the necessary representational groundwork upon which the complete CAD environment will be based: the PIF database provides the framework for the representation of the wafer, while the Process Flow Language (PFL) provides the representation of the process. In the last six months, these representations have been developed to the point where we were able to begin experimentation with tools based on the PIF and PFL.

We have developed an experimental Process Simulation Manager, and coupled this with the need to simulate the MIT Baseline CMOS process. The baseline process was represented in our experimental Process Flow Language. A nominal "Suprem-III Translator" produces fragments of Suprem-III code to simulate operations within the flow. Using UNIX utilities, particularly make, the prototype Manager enables us to maximize the sharing of computation between multiple cross sectional simulations, as well as minimize and automate resimulation when the process changes. A number of post-processing utilities allow the interactive analysis of final and intermediate simulated profiles.

The prototype Simulation Manager does not provide tight coupling between the flow and simulation, nor does it manage other important aspects of design besides simulation. We are continuing development of the Simulation Manager, as well as a Design Supervisor to provide additional capabilities in process verification, analysis, and synthesis.

EQUIPMENT MODELING

During this period the first equipment model was completed. This model concerns the low pressure chemical vapor deposition of polysilicon in a horizontal tube furnace. The model consists of a one dimensional finite difference numerical formulation which encompasses convective and diffusive mass transport in the annular space between the wafers and the tube liner, and accounts for the surface reaction rate limited deposition of polysilicon on the wafers with the associated generation of hydrogen and incorporation into the bulk of the gas. The model permits as input the gas flowrate to the three injectors, positions of the injectors, reactor geometry, temperature profile down the tube, and operating pressure. The adjustable coefficients in the model have been calibrated using a series of designed experiments performed at the applications lab of BTU Bruce of Billerica, Massachusetts. In these experiments, 150 wafer loads of six-inch wafers were used. The experiments involved four parameters, two gas flowrates, one injector position, and operating pressure.

The results from the model are in excellent agreement with the experimental work. The model appears to predict the profile of growth rate down the tube accurately. The model is accurate enough to be interrogated for process optimization, and gives a predicted optimum set of process parameters which is very close to that found by the experimental Taguchi optimization.

In the near future, the model will be tested in on-line quality control for its ability to predict changes around an operating point.

During this period, George Prueger has completed his master's thesis on LPCVD of poly. A paper is in preparation for submission to the Journal of Semiconductor Manufacturing.

Also during this period, Michele Storm has investigated the use of a software package called ULTRAMAX for on-line quality control in CAF.

MECHANICAL-PROPERTY TCAD

This is a new project, directed toward the development of Technology CAD (TCAD) tools with which to predict the mechanical behavior of microfabricated devices. The application of these tools will be twofold: prediction and modeling of microsensor and microactuator devices as part of the design process; and analysis of stress distributions in microelectronic parts for reliability assessment. There are many well-documented examples of device failure produced by mechanical failure (cracking of dielectrics and conductors, and delamination of coatings). At present, these are handled on a case-by-case basis, and only when the fault is detected during life test of finished parts. The goal here is to use research on test structures to build a data base and CAD environments with which to model the stress distributions in microelectronic structures prior to fabrication in order to identify high-risk sections of a design that might be prone to catastrophic mechanical failure. In addition, these tools, developed initially with the use of experiments on deformable structures, such as beams, cantilevers and diaphragms, will be used for predictive modeling during the design of sensors and actuators that include such deformable components. Effort to date has concentrated on developing data on the stress and modulus of polysilicon as a function of its processing history.

SCHEDULING

Research during this period focused on three activities: studying the integrated circuits fabrication process at a systems level; formulating a mathematical model of an integrated circuits fabrication facility; and *developing simulation* and scheduling software.

The effort to define the scheduling problem continues. We are concentrating on the MIT laboratories as case studies. Mathematical and simulation models, described below, will be based on what we learn here. A draft report has been written summarizing our knowledge of semiconductor fabrication from the point of view of a scheduler. We have sent it to many well-informed people in industry and academia and solicited their comments by including a questionnaire. Some comments have come back, but we expect to get the bulk of the responses during April 1988. Readers who would like to review this draft and possibly make comments are urged to contact us.

The multiple-time-scale decomposition is under development and shows great promise. A new basic model is being investigated which will help us to refine and better justify the tentative mathematical results we have developed thus far on hierarchical scheduling.

In this approach, the scheduling algorithm is divided into a set of levels which correspond to classes of events that are distinguished by their frequencies. At each level, two kinds of calculations are performed: small linear programs, to determine frequencies of higher frequency (lower level) events; and simple combinatorial optimizations, to determine exact times for the events of that level, whose *frequencies have been calculated at* higher levels.

We are devoting a great deal of attention to the scheduling of setups since they are likely to become important in modern multiple-purpose fabs, i.e., those that can be used for more than one basic process.

Software which will implement this multiple time scale decomposition approach to hierarchical scheduling is under development. A simulator is also under development. The scheduling software will first be tested with the simulation, and then used to run the laboratory.

PUBLICATIONS LIST

S. B. Gershwin, "A Hierarchical Framework for Discrete Event Scheduling in Manufacturing Systems," Springer-Verlag Lecture Notes in Control and Information Sciences, Vol. 103, Discrete Event Systems: Models and Applications, edited by P. Varaiya and A. B. Kurzhanski, pp. 197-216, August, 1987.

E. Sachs and G. Prueger, "Process Model Testing and Optimization Using Matrix Experimentation," Proceedings, ASME Winter Annual Meeting, December, 1987.

E. M. Sachs, G. Prueger, and R. Guerrieri, "Process Model Testing and Optimization Using Orthogonal Array Experimentation," presented at the 1988 Winter Annual Meeting American Society of Mechanical Engineers, Boston, MA, January, 1988, PED Volume 27.

T.-L. Tung, J. Connor, and D. A. Antoniadis, "A Boundary Element Method for Modeling Viscoelastic Flow in Thermal Oxidation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 7, no. 2, pp. 215-224, February, 1988.

E. M. Sachs and G. Prueger, "Process Model Construction and Optimization Using Statistical Experimental Design," Proceedings of Manufacturing International '88, Atlanta, GA, April, 1988. Also, MIT VLSI Memo No. 88-442, March, 1988.

D. S. Boning and D. A. Antoniadis, "A Workstation Approach to IC Process and Device Design," IEEE Design and Test of Computers, vol. 5, no. 2, pp. 36-47; April, 1988.

T. A. Lober, J. Huang, M. A. Schmidt, and S. D. Senturia, "Characterization of the Mechanism Producing Bending Moments in Polysilicon Micro-Cantilever Beams by Interferometric Deflection Measurements," to be presented at the 1988 Hilton Head Workshop on Solid-State Transducers, June, 1988.

O. Z. Maimon and S. B. Gershwin, "Dynamic Scheduling and Routing For Flexible Manufacturing Systems that have Unreliable Machines," to appear, Operations Research.

INTERNAL MEMORANDA

X. Bai and S. B. Gershwin, "A Manufacturing Scheduler's Perspective on Semiconductor Fabrication," in preparation.

TALKS WITHOUT PROCEEDINGS

D. E. Troxel, "Computer Aided Fabrication," Digital Equipment Corp., September 2, 1987.

S. B. Gershwin, "Control and Systems Issues in Manufacturing -- Some Problems and Solutions," MIT Laboratory for Manufacturing and Productivity, October 1, 1987.

S. B. Gershwin, "A Hierarchical Framework for Manufacturing Systems Scheduling," MIT Sloan School of Management, OR/OM Workshop, October 5, 1987.

P. Penfield, Jr., ' Computer-Aided Fabrication of Integrated Circuits," IEEE International Conference on Computer Design '87, Port Chester, NY, October 5, 1987.

S. B. Gershwin, "A Hierarchical Framework for Manufacturing Systems Scheduling and Planning," National Bureau of Standards, Gaithersburg, MD, November 2, 1987.

S. B. Gershwin, "A Hierarchical Framework for Manufacturing Systems Scheduling and Planning," University of Kentucky, November 19, 1987.

G. Prueger, E. Sachs, and R. Guerrieri, "Equipment Model for the Low Pressure Chemical Vapor Deposition of Polysilicon," MIT VLSI Research Review, Cambridge, MA, December 14, 1987.

S. B. Gershwin, "A Hierarchical Framework for Manufacturing Systems Scheduling and Planning," Rensselaer Polytechnic Institute, Troy, NY, March 24, 1988.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY VLSI PUBLICATIONS

VLSI Memo No. 88-442
March 1988

PROCESS MODEL CONSTRUCTION AND OPTIMIZATION USING STATISTICAL EXPERIMENTAL DESIGN

Emmanuel Sachs and George Prueger

Abstract

A methodology is presented for the construction of process models by the combination of physically based mechanistic modeling and statistical experimental design in order to create "smart" response surfaces. In contrast to the process independent polynomial fit of the conventional response surface method, smart response surfaces derive their basic shape from the process physics and are then calibrated using designed experiments. This method provides for a surface of better representational accuracy using the same or fewer experimental points.

This method has been applied to the development of a model for the low pressure chemical vapor deposition (LPCVD) of polysilicon, a process used in the manufacture of VLSI circuits. A one-dimensional finite difference model of the LPCVD process was constructed. A Taguchi orthogonal array experiment was conducted. A confirming experiment performed at the parameter levels indicated by the Taguchi optimization, served to confirm the validity of the experimental procedure. The experimental results will subsequently be used to calibrate the mechanistic model.

Microsystems
Research Center
Room 39-321

Massachusetts
Institute
of Technology

Cambridge
Massachusetts
02139

Telephone
617 253-8138

ABSTRACT

A methodology is presented for the construction of process models by the combination of physically based mechanistic modeling and statistical experimental design in order to create "smart" response surfaces. In contrast to the process independent polynomial fit of the conventional response surface method, smart response surfaces derive their basic shape from the process physics and are then calibrated using designed experiments. This method provides for a surface of better representational accuracy using the same or fewer experimental points.

This method has been applied to the development of a model for the low pressure chemical vapor deposition (LPCVD) of polysilicon, a process used in the manufacture of VLSI circuits. A one-dimensional finite difference model of the LPCVD process was constructed. A Taguchi orthogonal array experiment was conducted. A confirming experiment performed at the parameter levels indicated by the Taguchi optimization, served to confirm the validity of the experimental procedure. The experimental results will subsequently be used to calibrate the mechanistic model.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency under contract N00014-85-K-0213 as well as the support of the Microelectronics and Computer Technology Corporation.

The authors would like to thank BTU Engineering Corporation with special thanks to Mr. Arthur Waugh and Mr. Morris Simson for providing advice, guidance, and access to experimental facilities.

MOTIVATION FOR PROCESS MODELING

In the most general sense, a process model is a body of knowledge which provides predictions about the outputs from a manufacturing process given information about inputs to the process. Figure 1 illustrates a generic manufacturing process and related processing equipment. The inputs have been divided into usefully distinct categories; process parameters or control factors and noise factors. The process parameters are those parameters that we exercise direct control over. Examples of process parameters include temperature, pressures, roll speeds, gas flow rates, etc. A second set of inputs is entitled disturbances or noise factors. Noise factors are inputs to the process which are subject to unintended and undesired variation. Examples of noise factors include variations in the properties of incoming raw material, and in the process parameters themselves. The goal of the process model is to provide information about the output from the process given information about the process parameters and the noise factors.

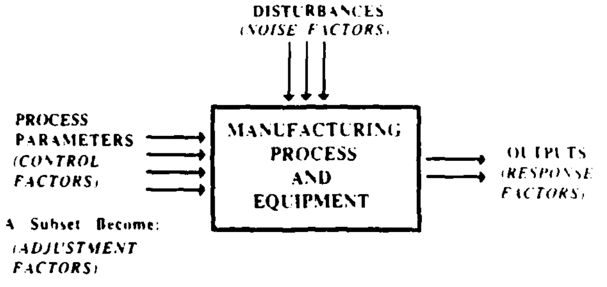


Figure 1. Representation of a Generic Manufacturing Process.

Process modeling has several critical roles to play. A competent model of a manufacturing process is an extremely powerful tool in the design of new processing equipment, as accurate predictive capability will substantially reduce the number of iterations necessary to achieve a satisfactory design.

Process modeling is also essential to the operation of existing equipment. Models can be used to optimize the operation of existing equipment. Optimization might be accomplished by selection of a set of process parameters which leads to the greatest robustness of the process against disturbances or noise factors. Process models are also extremely useful for on-line quality control. While we may think of process optimization as the selection of an operating point for the process, on-line quality control deals with process operation around that operating point. Models can be used to effectively guide the process back to target values.

The construction and utilization of process models is especially critical in modern manufacturing environments. Competitive pressures dictate that processes must be run near their optimum conditions. Computer integrated manufacturing offers a potential wealth of data from process operations which can only be effectively utilized in combination with process models.

APPROACHES TO MODEL CONSTRUCTION

Models may be constructed by three distinctly different approaches; experimental, experiential, and analytical.

Experimentally based process models are constructed by performing deliberate and planned experiments on a process. These experiments are most effectively performed using techniques of statistical experimental design. In these methods such as Box "factorial experimental design and response surfaces" [1, 2] and Taguchi "orthogonal array" [3, 4, 5, 6], many experimental parameters are varied simultaneously in a well-defined plan, resulting in great economy of experimentation. These methods are reviewed briefly in a later portion of this paper.

Experiential process models may be constructed simply by operating the process in production and collecting and analyzing the "observational" or "happenstance" data that results.

Analytical models are based on the fundamental physical mechanisms of the problem. Analytical models can be either closed form, or numerical methods

Physically based analytical models offer the advantage of having the greatest generality and range of application, however, they can be extremely time consuming to develop, and are often of questionable accuracy and use because of a lack of complete knowledge about the process physics. Experimental and experiential models offer the advantage of good fidelity within the range of variables tested but limited extrapolation capacity beyond that range and limited extension to equipment other than that upon which the experiments were run.

In today's practice, the three methods of model building, experimental, experiential, and analytical, are generally applied independently with little interaction between the methodologies. This papers concerns the fusion of analytical and experimental modeling in order to gain the generality of an analytical model in combination with the precision and ease of use of an experimental model.

BACKGROUND -- DESIGNED EXPERIMENTS

Experimental design is a systematic and organized way to conduct experiments in order to extract the maximum information from the minimum number of experiments. The unifying feature of statistically designed experiments is that all the parameters of interest are varied simultaneously, in contrast to the more conventional one variable at a time experimental technique. In this manner, the total experimental range is explored with a minimum number of experiments. There are two commonly used methodologies for experimental design; the Taguchi orthogonal array method and the Box response surface method.

In the Taguchi orthogonal array method, scientific and engineering knowledge is used to pick experimental parameters which are non-interacting. Typically, three levels would be assigned for each parameter, low, medium and high. The parameters or "control factors" are then arranged in an orthogonal experimental array. Figure 2 shows a four parameter, three level, orthogonal array which defines nine experiments. Also shown in Figure 2 is a graphical representation of the distribution of experimental points for three levels of three control factors or parameters. The points plotted correspond to the second, third and fourth columns of the orthogonal array shown. Such a plot is useful in visualizing the distribution of experimental points in space, but loses its utility past three control factors. The unique feature of an orthogonal array is that for a given level of a given parameter, all other levels of all other parameters are explored uniformly. Thus, for example, in runs one, two and three of the orthogonal array in Figure 2, parameters two, three and four are all rotated through their low, medium and high values.

	Parameters			
Runs	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

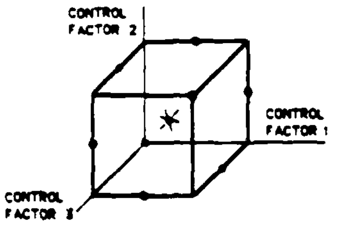
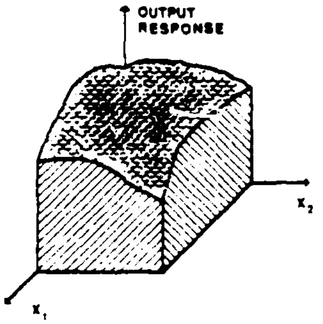


Figure 2. A Taguchi Orthogonal Array and 3 dimensional visualization of columns 2, 3 and 4.

Another unique feature in the Taguchi method is the simultaneous use of "inner" and "outer" arrays. The array shown in Figure 2 is an inner array, and is used to define the points of interest in experimental space. An outer array would be used to define small regions around each of the points specified in the inner array. The outer array would specify, for example, variations or tolerances on experimental parameters or other "noise" factors, thereby defining additional experiments in the neighborhood of each of the nine experiments specified by the inner array.

Perhaps the most distinct feature of the Taguchi method is the interpretation of results by using a "signal to noise ratio". This method of interpretation will be discussed in greater detail later in the paper.

The alternative method, developed by George Box and others, defines a series of experiments and summarizes the results of those experiments in the form of a response surface. A response surface is a polynomial fit (usually a quadratic polynomial) to the measured data. The concept of a response surface and its analytical representation is shown in Figure 3 for a function of two variables, x_1 and x_2. Since the response surface includes the effect of factor interactions, a larger number of experiments is needed to fit a response surface as compared to the Taguchi orthogonal array method. For example, a second degree polynomial fitted to four parameters at three levels, would require a minimum of fifteen experiments as opposed to the nine experiments used in the Taguchi method.



$$\text{RESPONSE} = C_1 + C_2X_1 + C_3X_2 + C_4X_1^2 + C_5X_2^2 + C_6X_1X_2 + \ldots$$

Figure 3. An Illustration of a Box Response Surface.

The response surface method is powerful in its generality, but suffers from the fact that it does not directly embody the sensitivity of the output or response to small deviations of input factors. Since the response surface is a quadratic fit to three points, one cannot expect that the slope at the three points is particularly accurate. Since it is the local slopes that embody the sensitivity information which will be necessary to design a robust process, the quadratic fit response surface is not particularly useful for the design of robust processes.

BUILDING "SMART" RESPONSE SURFACES

The basic conceptual framework of the present work is to replace the polynomial fit of a response surface with a shape dictated by the physics of the process under study. Since the general shape will be dictated by the physics, a more precise model can be obtained with the same number of experimental points, and perhaps even with fewer experimental points. In addition to being more accurate within the experimental range, such a model would also be more useful when extrapolated beyond the tested range.

The remainder of this paper describes one method of constructing such a "smart" response surface. The steps in construction are as follows:

- identify and characterize parameters or factors
- perform designed experiments
- develop a simple analytical model
- calibrate the model
- use the model for a design and operation of equipment

BACKGROUND -- LOW PRESSURE CHEMICAL VAPOR DEPOSITION OF POLYSILICON

The process modeled in this paper is a low pressure chemical vapor deposition process used in the semiconductor industry to fabricate VLSI circuits. Integrated circuits are basically fabricated by alternately depositing and selectively removing layers on a silicon wafer. In the dominant family of CMOS circuits, an individual transistor has three contacts; the source, drain, and gate, as shown in Figure 4 [7]. The most commonly used material for the gate electrode, is polycrystalline silicon, which is deposited on the wafer approximately midway through the fabrication of a CMOS circuit. Polysilicon is used as a conductor at this intermediate fabrication stage, because it allows the wafer to be exposed to subsequent high temperature processing which metallic contacts would preclude.

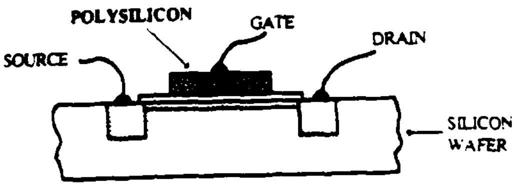


Figure 4. Schematic Cross Section of a MOSFET Transistor.

Polysilicon layers are most commonly deposited on wafers in a batch operation in a tube furnace as shown in Figure 5 [8, 9]. A tube furnace typically consists of a quartz tube surrounded by resisting heating coils which are in turn surrounded by thermal insulation. The four to six inch silicon wafers are held 25 at a time in "boats". Four to six boats are loaded into the tube furnace for each batch. After loading, a partial vacuum is applied to the tube and a process gas in introduced through small tubular "injectors". For low pressure chemical vapor deposition (LPCVD) of polysilicon, the process gas is SiH_4. At the operating temperature of approximately 625° C., SiH_4 pyrolytically decomposes to yield solid silicon which deposits on all hot surfaces, and gaseous hydrogen, according to the relationship:

$$SiH_4 \xrightarrow{625^\circ C.} Si + 2H_2$$

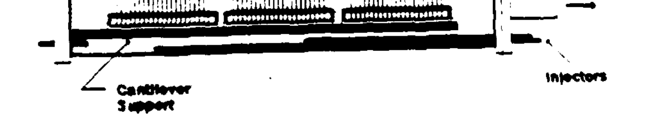


Figure 5. A tube furnace used for the deposition of polycrystalline silicon.

As the reaction proceeds, the liberated hydrogen joins the process gas in the tube, thus diluting the mixture and reducing the reaction rate. It is primarily this dilution which leads to non-uniformities in deposition or growth rate from wafer to wafer down the length of the tube. Wafer to wafer non-uniformity is the primary source of concern in LPCVD of polysilicon. For reasons explained in detail below, across the wafer uniformity tends to be quite good in this process. In order to counteract the SiH_4 depletion and improve the wafer to wafer uniformity in the tube, SiH_4 is introduced or "injected" at three sites in a typical tube furnace as indicated schematically in Figure 5.

Process control is achieved by adjustment of the flowrates through each of the three injectors, adjustment of the axial position of the injectors, temperature adjustment, and control of tube pressure.

IDENTIFICATION AND CATEGORIZATION OF PARAMETERS

The primary control factors or parameters are tube pressure, flowrate through each injector, position of each injector, and temperature profile down the tube.

The possible list of noise factors is quite long, but might prominently include such issues as the amount of prior deposition in the tube, the prior condition of the wafers, aging of the control thermocouples, spacing between the wafers, and location accuracy of the wafers in the tube.

The output or response factor chosen for the current model is the deposition thickness as a function of position down the tube. In practice, one is also concerned with controlling the grain size of the deposition, and the thickness distribution across each wafer. As noted earlier, the thickness across a wafer tends to be quite uniform. As the grain size is controlled almost exclusively by deposition temperature, the desired grain size fixes the temperature of deposition typically at 625° C.

For the purposes of our modeling, we have chosen the four control factors: tube pressure, the flowrate through the load end injector, the flowrate through the center injector, and the position of the source injector. The load end injector is the injector on the left side of Figure 5. The source end injector is the injector whose opening within the tube furnace is furthest to the right in Figure 5. The output or response factor in our experiment will be the profile of thickness down the length of the tube furnace.

EXPERIMENTAL DESIGN, RESULTS, AND INTERPRETATION

The experimental design used was a Taguchi orthogonal array using four parameters at three levels. This array, shown in general form in Figure 2, is again shown in Figure 6 complete with parameter assignments and level selections. Three of the four parameters (all but pressure) are indicated in dimensionless form. The nine experiments were conducted with a wafer load of 150 six-inch wafers. Thirteen test wafers were distributed within the 150 wafer load. A baseline experiment was repeated five times in order to gain some information about run to run variability of the process.

Figure 7 shows plots of growth rate (averaged over the test wafer) against position in the tube furnace for runs one and nine of the orthogonal array. These plots are indicative of the range of results obtained. The mean values and standard deviations of each of the thirteen test wafer positions are shown. The mean value has been obtained from a single run of each experiment. The standard deviation was obtained by normalizing the standard deviation for each wafer position in the baseline replicate runs and applying this normalized standard deviation to the nine Taguchi array experiments. The bar charts at the bottom of the plots in Figure 7 schematically indicates the gas flow at each injector site. As may be seen, there is a general correlation between local injector volume and growth rate. The system thus seems to have no conspicuous pathologies.

Experiment Number	Pressure (mtorr)	Q_{load} (% of total)	Q_{center} (% of total)	X_{source} (% of tube length from center)
1	200	20	26.7	9
2	200	30	36.7	12
3	200	40	46.7	15
4	250	20	36.7	15
5	250	30	46.7	9
6	250	40	26.7	12
7	350	20	46.7	12
8	350	30	26.7	15
9	350	40	36.7	9

Figure 6. Taguchi Orthogonal Array Experimental design used. $Q_{load} + Q_{center} + Q_{source} = 150$ stnd cm^3/min. Process temperature = 625°.

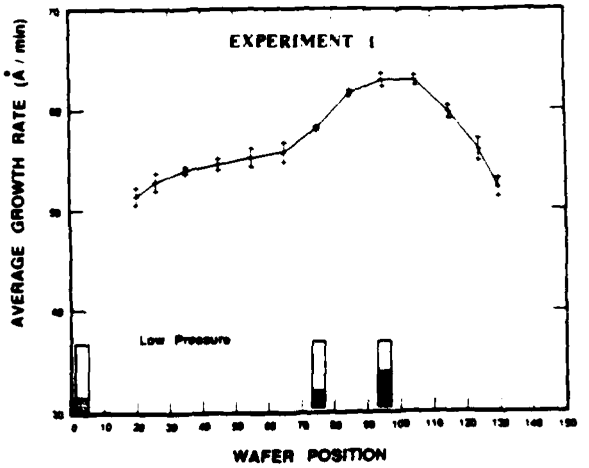
The results of the designed experiments are to be used in two ways. Later they will be used to calibrate the smart response surface model. First, however, they will be used to predict an "optimized" point of operation of the equipment. This predicted optimum will then be run in a confirming experiment. If the resulting improvement is close to that predicted, we may be confident that the experimental parameters and output variables have been properly identified and that the experiments were performed well.

The interpretation of the orthogonal array results is indicated schematically in Figure 8, and begins by calculation of a signal to noise ratio for each of the nine experimental runs. This signal to noise ratio characterizes the deviation of each of the profiles from a flat and uniform profile. Next, average signal to noise ratios are calculated for each level of each parameter, and are plotted on the marginal graphs of Figure 9. These graphs may now be used to select the best combination of parameter levels for an optimized process. This optimum occurs at middle values for load injector, flow source injector, low values for flow source injector position and pressure.

This combination of parameters may now be run in a confirming experiment, the results of which are shown in Figure 10. As may be seen, the confirming experiment is a substantial improvement, thus validating the experimental procedure.

DEVELOPMENT OF THE ANALYTICAL MODEL

The development of the analytical model begins by performing order of magnitude calculations in order to identify important physical mechanisms. First, mass transport calculations indicate that the the deposition rate is limited by the surface reaction rate, and not by transport to the surface. The flow in the annular region between wafers and tube is characterized by a Reynold's number of approximately one, indicating that the flow is laminar. The Peclet number in the annular region is approximately one, indicating that convective and diffusive fluxes are roughly comparable, and that both must be considered in the solution to the problem. A Poiselle flow calculation in the annular space indicates that the pressure drop down the length of the tube is less than one percent of the actual pressure in the tube, and hence is negligible.



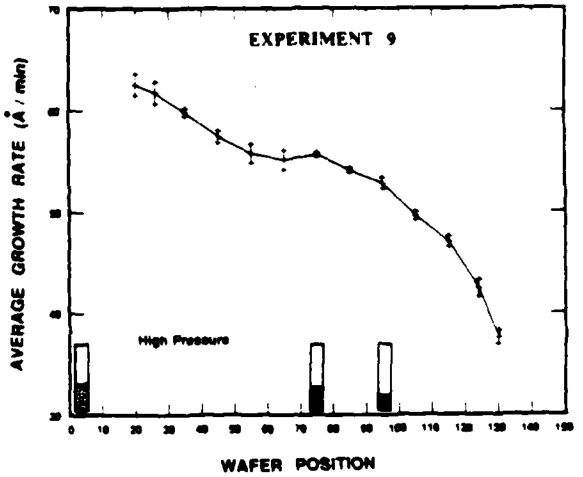
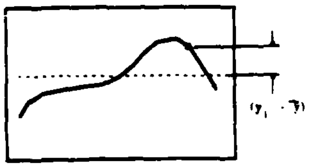


Figure 7. Measured growth rate plotted against temperature for two of the nine experiments defined in Figure 6.

In the region between wafers, the Peclet number is much less than one, indicating that diffusive fluxes dominate, and that the convective flow in this region can be ignored. The exit velocity from the small diameter injector tubes is on the order of 0.5 Mach 1, and hence, the flow geometry and incorporation into the tube annular space is quite complex. The gas depletion, that is the percentage of SiH_4 that is reacted to form silicon, is between 20% and 50%, and is thus, an important component of the problem. Indeed, it is this depletion that is the primary source of the variation in deposition rate. The surface reaction rate is a function of the local partial pressures of SiH_4 and hydrogen and the local temperature.

Consistent with these order of magnitude calculation, our model assumes a one dimensional finite difference formulation with no radial non-uniformities. The flow has been assumed to be inviscid, and to be of a plug flow nature. We are concerned with the coupled problem of convection and diffusion in the annular space. We will consider the thermal problem to be separated from the flow problem and will specify the wafer temperature as a function of position down the tube. We will also specify the annular flow area as a function of position down the tube. We will model the injector flow as mixing into the annular space in the tube over a tube length that is proportional to the square of the flowrate (and velocity) in the tube.

	Parameters				
Runs	1	2	3	4	S/N
1	1	1	1	1	23.01
2	1	2	2	2	27.55
3	1	3	3	3	18.61
4	2	1	2	3	21.91
5	2	2	3	1	20.89
6	2	3	1	2	23.24
7	3	1	3	2	24.23
8	3	2	1	3	24.79
9	3	3	2	1	17.30



$$S/N = 10 \log_{10}\left(\frac{\text{Mean}^2}{\text{Variance}}\right)$$

$$\text{Variance} = \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n - 1}$$

Figure 8. The experimental results are interpreted by calculating a signal to noise ratio for each of the nine experiments of the orthogonal array.

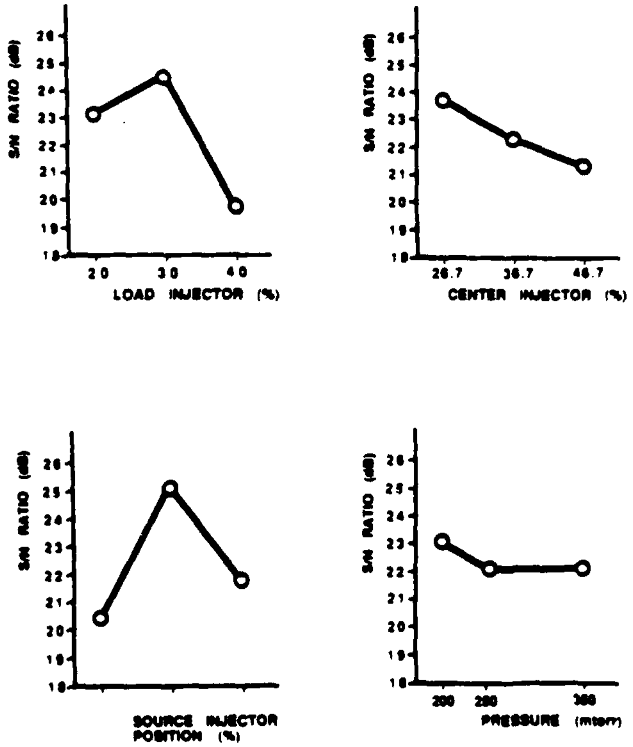


Figure 9. The "marginal graphs" which plot average signal/noise ratio for each level of each parameter.

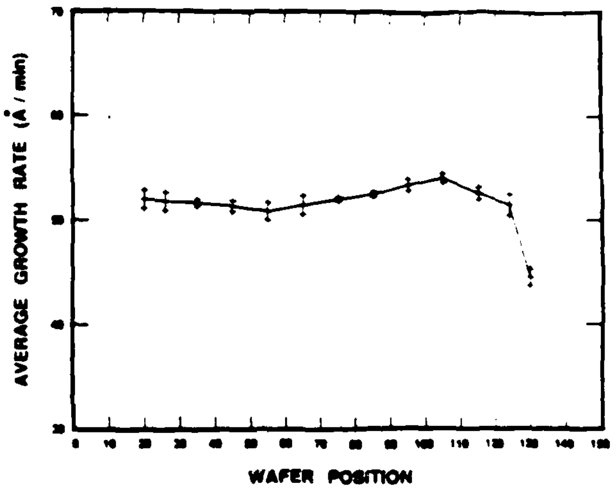


Figure 10. Confirming experiment at "optimized" control factors.

The model basically consists of mass conservation conditions for each of the two species, SiH_4 and hydrogen taken individually and together as summarized in the equations below.

Mass Conservation of Silane:

$$\frac{d}{dz}\left(AcD\frac{dx_s}{dz} - AcV\,x_s \right) = R(x_s) - F$$

Mass Conservation of both species:

$$\frac{d}{dz}(c\,A\,V) = R(x_s) + F$$

where:

A is the cross sectional area of the flow [cm^2],
C is the total concentration of gaseous species [$moles/cm^3$],
z is the distance down the tube [cm],
D is the diffusion of silane [cm^2/sec],
x_s is the molar fraction of silane [moles/mole],
V is the molar average velocity [cm/sec],
R is the surface reaction rate and is a function of x_s and temperature, [mole/sec · cm],
F represents the injector inlet flow [mole/sec · cm]

MODEL CALIBRATION

The final step in construction of a smart response surface is the use of the results of the experimental design to perform a least squares fit to calibrate the model parameters. The adjustable parameters in our model include four parameters that govern the deposition rate as a function of partial pressures and temperature, and two parameters that specify the nature of the incorporation of injector flow into the annular space.

At the current writing, the model has been developed and found to converge to proper solutions rapidly. In the near future, the model will be calibrated using the experimental data.

CONCLUSIONS AND FUTURE WORK

We have illustrated a methodology for the construction of process models using the concept of the "smart" response surface. A smart response surface is one in which the general shape of the response surface is dictated by the process physics and the shape is then calibrated using statistically designed experiments. This approach contrasts with the conventional response surface methods in which a general polynomial is fit to the data derived by designed experiments. The motivation behind constructing a smart response surface is that the resulting model will have greater fidelity in representing the process both within the range of experimentation and when extrapolated beyond the experimental range.

The smart response surface approach has been discussed within the context of building a model for the low pressure chemical vapor deposition of polysilicon as performed in the integrated circuit industry. In the work discussed, a finite difference model of the process was built, and a Taguchi orthogonal array experimental design was performed. The results from the designed experiment were used to optimize the process and a confirming experiment at the predicted optimum conditions demonstrated the validity of the experimental program. In the future, the designed experiments will be used to calibrate the numerical model.

Future work will focus on ways to combine designed experiments with fragmentary mechanistic modeling. This will permit the use of the smart response surface technique without the requirement that a complete physical model be available.

REFERENCES

1. Box, G., Hunter, W., Hunter, J., Statistics for Experimenters, and Introduction to Design Data Analysis and Model Building, Wiley, 1978.

2. Box, G., Draper, N., Empirical Model-Building and Response Surfaces, Wiley, 1987.

3. Taguchi, G., Introduction to Quality Engineering, Krauss International Publications, White Plains, New York 1986.

4. Taguchi, G. System of Experimental Design: Engineering Methods to Optimize and Minimize Costs Vol. 1, Kraus International Publications, White Plains, NY.

5. Phadke, M., Kucjar, N., Speeney, D., and Grieco, M., "Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental Design". The Bell System Technical Journal, 1983.

6. Nackar, R., Shoemaker, A., "Robust Design: A Cost-Effective Method for Improving Manufacturing Processes", AT&T Technical Journal, March/April 1986.

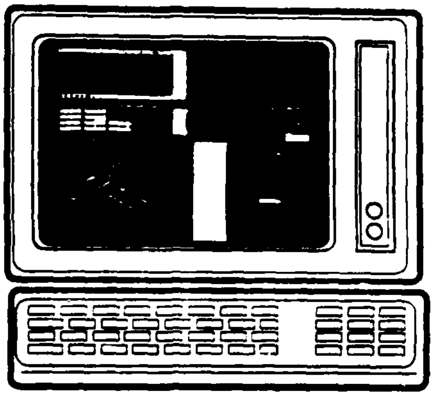
7. Senturia, Stephen D., Wedlock, Bruce D., Electronic Circuits and Applications, John Wiley & Sons, New York.

8. Wolf, Stanley, and Tauber, Richard, Silicon Processing for The VLSI Era, Volume 1: Process Technology, Lattice Press, Sunset Beach, California.

9. Sze, S. M., VLSI Technology, McGraw-Hill Book Company, New York.

10. Jensen, K.F. and Graves, D.B., "Modeling and Analysis of Low Pressure CVD Reactors", J. Electrochem Soc.: Solid-State Science and Technology, September 1983.

11. Van Den Brekel, C.H. and Bollen, L.J.M. "Low Pressure Deposition of Polycrystalline Silicon from Silane", Journal of Crystal Growth 54, 1981.



A WORKSTATION APPROACH TO **IC PROCESS AND DEVICE DESIGN**

DUANE S. BONING
DIMITRI A. ANTONIADIS
Massachusetts Institute of Technology

IC designers are turning more and more to CAD tools to develop complex designs and to automate time-consuming tasks. Although there are a variety of integrated tools for many types of VLSI design, very few integrated systems have been built to address process and device design. Recognizing this need, researchers at MIT set out to define the requirements of a process and device design environment, implement a subset of these functions, and integrate the tools into a user-friendly design environment. As part of their work on creating a user-friendly environment, they developed the MASTIF workstation to provide graphic, window-oriented user interaction to process and device designers.

Spurred by the increasing complexity of ICs, developers are providing CAD tools to help in the many phases of IC design. These tools, which span the architectural, system, logic, circuit, device, and process levels of design, can be integrated to provide a tool for successive design phases. Design in several of these domains can be done on workstations, in which tools for specification, capture, and synthesis are put together with verification, simulation, and analysis programs.

The design of semiconductor devices and fabrication processes, on the other hand, suffers from the lack of a similar integrated workstation approach. A limited number of tools do provide simulation capability: process simulators such as Suprem-III from Stanford University[1] model aspects of the fabrication process, and device simulators such as Minimos[2] evaluate MOS and other semiconductor devices. Nevertheless, process and device design has long been the poor cousin in the VLSI CAD family. Only a very few integrated systems provide a full complement of tools—particularly tools beyond simulation—in device and process design.

If process and device tools are to evolve in the same way as other VLSI CAD tools, then it follows that the same types of abstract capabilities will be needed. Through our work at MIT, we have been able to define the requirements

0740-7475/88/0400-36$1.00©1988

and composition of a complete process and device design system that we feel can serve as a blueprint for the future evolution of process and device design tools. As part of our research, we developed the MASTIF (short for MIT analysis and synthesis tool for IC fabrication) workstation, which meets some needs for process and device design.

CREATING A SYSTEM

The evolution of CAD tools for VLSI design follows a recurrent pattern. Under the pressures of increasing model size and complexity, simple hand and paper methods grow into sophisticated CAD tools with specification, synthesis, capture, verification, simulaticn, and analysis. These, in turn, are integrated with other CAD tools.[3] Device and process design tools have not yet reached the degree of development characteristic of other areas in VLSI design. For the most part, process design is done with a variety of process or device simulators and only limited user-program and program-program interfacing.[4] Few systems integrate these simulation tools *and* provide tools that are comparable to those available in other VLSI design areas.[5]

As mentioned earlier, we believe that process and device design will follow the same evolution as other areas of design. Hence, we also believe that an integrated process and device design system is needed to aid in specification, synthesis, capture, verification, simulation, and analysis. As shown in Figure 1, these functions are needed in both process and device design.

The functional needs for process and device design shown in the figure are by no means exhaustive. An environment providing these capabilities, however, would be extremely useful in the design of a baseline device or process. Additional tools to support the realistic design of devices and processes will evolve as they are needed. For instance, the process and device analysis functions will grow to include tools for variational or yield analysis; the synthesis function will grow to include design centering as well as optimization capabilities.

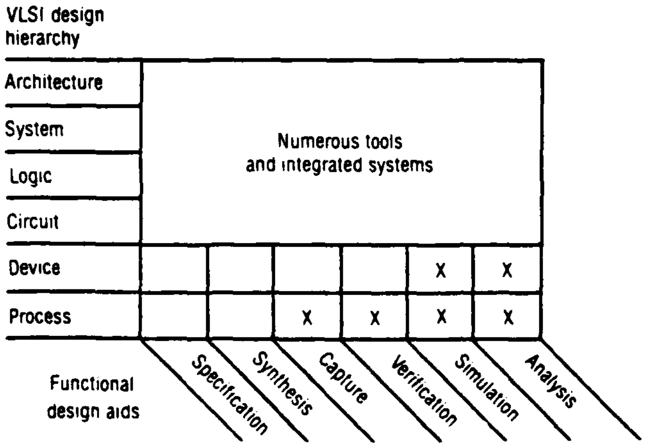
The framework outlined here is intended only as a guide both to the development of individual process and device design tools and to the integration of these tools into a complete design environment.

SPECIFICATIONS

Creating a design begins with developing specifications. The engineer is striving to satisfy a large number of goals or requirements, which may be formal or informal. Process specification is the task of enumerating the various goals, requirements, and constraints on the fabrication process or fabricated structure. Device specification is the task of defining the electrical and structural characteristics of a completed semiconductor device.

Process specification poses the bigger problem. Various specifications may be needed, such as the thickness and composition of layers, characteristics of two-dimensional topological features (the extent of lateral oxidation, for instance), sheet resistance of diffused regions, and junction depth requirements. To date, there are no facilities for formally specifying process requirements, primarily because there is no design automation capability that could support them.

Device designers are more familiar with electrical specificatıons. Because a device's electrical performance depends so strongly on the device's structure itself, we find that process design and device design are tightly coupled. As a result, we may need to merge process and device specification functions. Both types of specifications are required before we can make any substantial progress toward automated process and device synthesis.



VLSI design hierarchy	Specification	Synthesis	Capture	Verification	Simulation	Analysis
Architecture	Numerous tools and integrated systems					
System						
Logıc						
Circuıt						
Devıce					X	X
Process			X	X	X	X
Functional design aıds						

Figure 1. VLSI design hierarchy and functional design aids. The functions ma:kcd with X are included in the MASTIF workstation.

SYNTHESIS

In process design, synthesis serves as the bridge between a process specification and a process description. In device design, it serves as the bridge between a high-level expression of the device—the device specification—and an actual representation of the device, such as that an engineer might construct during device capture.

Again, the problems start with process design. In VLSI design, silicon compilation is almost entirely automated. The ultimate goal of process synthesis is also to automatically generate a fabrication process that meets specifications. However, automated process synthesis is not currently practical. Not only are facilities for process specification completely lacking, but the engineering trade-offs and dependencies in process and device design are often not completely understood. We need a better understanding of process and device design before we can build intelligent tools for process synthesis.

The device synthesis function can be viewed as the global purpose of the device or process engineer as well as of the design environment itself. Like process synthesis, device synthesis becomes a huge, unwieldy task. One way around this problem is to provide limited tools that will help the engineer synthesize the design manually. One such tool would optimize some measure of the design as a function of process parameters.

Synthesis tools might also include management aids for the engineer to control the design as it develops. Design documentation and version management facilities are examples of these types of tools.

CAPTURE

Process capture is a means of entering a fabrication process, somewhat like a schematic is captured in circuit design. For process capture, we first require a process description, or a representation (language) for expressing the process.[6] As we will see later, this problem—in the face of multiple needs like simulation, documentation, fabrication automation—is a difficult one.

The next thing we need in process capture is a tool that enables the engineer to interactively and incrementally enter and edit the process description throughout development.

In device capture, we make the interface to process design explicit. The device consists necessarily of semiconductor structures, the representation of which we get from process simulation. Engineers need a facility for capturing this device structure. Such a facility would allow them either to construct a device representation directly or to glue together existing profile representations. To provide this facility, we must standardize the way that the device structure and information about the structure are represented.

VERIFICATION

Process verification has two steps. First, we check the process description itself against sets of rules. We can validate the process description syntax by checking syntactic rules such as "you must always specify the time and temperature for a furnace step." We can check process description semantics in a way analogous to design rule checking in IC layout. Before simulation, we might verify that the process will produce a valid device structure. Finally, before we generate the "recipe" for a process, we can examine the process description to verify that it satisfies laboratory or fabrication area guidelines.

The second part of process verification is to check the finished (or intermediate) simulated structure against the initial process specification. This verification would be useful both as part of automated synthesis and as a separate utility available to the engineer.

Like process verification, device verification may have several parts. We might check the device representation before using it in very time consuming device simulations. We might also want to compare the simulated electrical characteristics with the device specification to verify that design requirements have been satisfied. Finally, we might compare simulated devices with measurements from fabricated devices to verify that the design is valid and that the simulated fabrication and the actual fabrication are in line.

SIMULATION

A process simulator models the effects of a fabrication sequence on a wafer. By performing successive simulations to model the sequence of fabrication operations, we can construct one-, two-, or even three-dimensional models of a device or wafer structure.[7] Thus, Suprem-III provides material and impurity concentration information for one-dimensional cross-sections, while UC Berkeley's Sample[8] calculates two-dimensional geometric effects from the lithographic, deposition, and etching process steps. Three-dimensional simulators are still in the experimental stage.

Process simulators typically manipulate or produce a wafer structure or wafer profile. This profile may represent not only the impurity concentrations and material composition in the silicon and other layers, but also the topological structure at the surface of the wafer. To date, these representations are peculiar to each simulator. Since we do not have a standard format for profile interchange, we need explicit interfaces between process and device simulators.

In addition to this type of interface, we need an interface between the engineer and the process simulator. Each simulator has its own input language for expressing the process. Creating these input files (particularly when using multiple simulators or simulating several cross sections of the wafer) is a repetitive and error-prone task. Just as we can generate inputs to circuit simulators from schematic representations, so we can construct inputs to process simulators automatically from the process description.

Finally, process simulation typically has large computation requirements. In addition to increased speed in simulation programs, other mechanisms are needed to reduce the amount of time spent performing simulations during process development. A multilevel simulation environment would be useful, for example. In the early stage of a design, simple, computationally inexpensive simulations might be sufficient. Later, we can perform more complete and complex simulations for better accuracy.

Device simulators calculate the electrical characteristics of a device structure in response to environmental conditions, such as temperature or bias conditions. For example, the threshold voltage of an MOS device or a response to a specified bias condition may actually be the success measure of a fabrication process and device. As in process simulation, we need to limit the tasks of the device simulator to just the calculation of these device characteristics. The device simulator should not have to provide graphical output, for example.

ANALYSIS

A simulated profile, of course, is useful in process design only if facilities are available for the examination and evaluation of that profile. Simple analysis capabilities, such as sheet resistance or junction depth calculations, are often built into process simulators. Such analysis should not be the job of process simulation tools. Instead, a powerful, general tool could analyze process and device information. Such a tool will be feasible when we arrive at a standard format for profile interchange.

We also need thorough analysis tools to evaluate the results of device simulation. We can use a general post-processor to examine simulation results interactively, for both text and graphics. Both device and process analysis tools should go beyond the simple presentation of results to include data reduction and manipulation.

MASTIF: AN INTERACTIVE TOOL

The MASTIF workstation is a first-generation attempt at providing some of the functions just discussed. Developed specifically for process and device design, MASTIF integrates many tools—both those we developed and those available elsewhere. The workstation is highly interactive: users work with a single graphics screen via a tablet and a keyboard. A variety of menus and windows are displayed and are available simultaneously.

Figure 2 shows a typical MASTIF screen, which consists of a process description window, a cross-section summary window, process simulation windows, a Suprem-III plot window, device simulation windows, a Midas window, and other interactive windows.

PROCESS DESCRIPTION

The process description window allows the engineer to interactively create and edit a fabrication process, which provides the process capture function. This process description is independent of any particular process simulator and, in fact contains no simulation directives. That is, the process description contains only the process as it applies to the entire wafer. It does not give a particular simulator's view or a cross-sectional view of that process.

In addition to statements for specifying each process step, the process description window includes constructs for managing versions and version branches of the process description. Table 1 lists statements that are currently part of MASTIF's process description function.

To open the process description window, the user enters a process step by typing the name of that step, such as "implant," followed by parameter names and parameter values, such as "arsenic dose=1e15 energy=100." The syntax for that step is checked immediately to ensure that the step has been entered correctly.

Figure 3a shows the process description in the block mode, in which only the name for each step is given, such as IMPLANT (using example just given). The block mode provides an overview of the process, and shows the different versions that the engineer has tried during development. MASTIF can also display the same process in sentence mode (Figure 3b), in which the text of the process step, including parameter values, is given.

CROSS-SECTION SUMMARY

The cross-section summary window (Figure 4) captures specific mask settings for each distinct cross section, and thus serves as the connection to VLSI layout information. The user enters the masking information in text format, and the mask definitions for the cross sections are displayed. MASTIF currently handles one-dimensional cross sections, differentiating two cross sections at the exposure step. Figure 5 shows the cross sections that are displayed in Figure 4 as representative of a simple NMOS process. In cross section 1, the mask setting is closed (resist is over oxide), while in cross section 2, the mask setting is open (no resist over oxide).

The process description window in Figure 3 and the simulation window in Figure 6 correspond to the cross-section summary window in Figure 4.

Table 1. MASTIF process description statements. Statements in Suprem-III are listed for comparison (note that comparable statements are not always available).

Statement Type	MASTIF	Suprem-III
Fabrication statements	Furnace	Diffusion
	Deposition	Deposition
	Epitaxy	Epitaxy
	Etch	Etch
	Expose	—
	Implant	Implant
	Initialize	Initialize
	Strip	—
Design documentation statements	Comment	Comment
	Title	Title
	$See.version	—
	$Version	—

PROCESS SIMULATION

The process simulation windows link existing process simulators and the MASTIF process description. Given the overall process description and specific cross-section masking information, MASTIF can create a simulation window for a particular simulator and update it automatically. So far, we have implemented only a Suprem-III simulation window. From the menus of these windows, the user can direct background simulation of particular steps.

The color of each step in the simulation window portrays the simulation status of that step. Red indicates an unsimulated step, yellow that the step is currently simulating in the background, and green that a step has successfully completed simulation. A neutral color is used for version blocks and stop blocks. The user can evaluate each simulated step independently by examining its simulated structure interactively.

Figure 6 shows the drain section simulation window specified in the cross-section summary window of Figure 5. In the first column, the first three blocks (medium grey) have successfully completed simulation. The same is true of the four blocks after $SEE.VER. The DIFFUSION and SAVE blocks (light grey) are currently simulating in the background. In the third column, the four blocks after $VERSION (dark grey) are unsimulated.

The window contains a valid input file to the Suprem-III simulator. The MASTIF translator, not the user, produces the file depicted in the process simulation windows, to make sure that translation among cross sections is consistent.

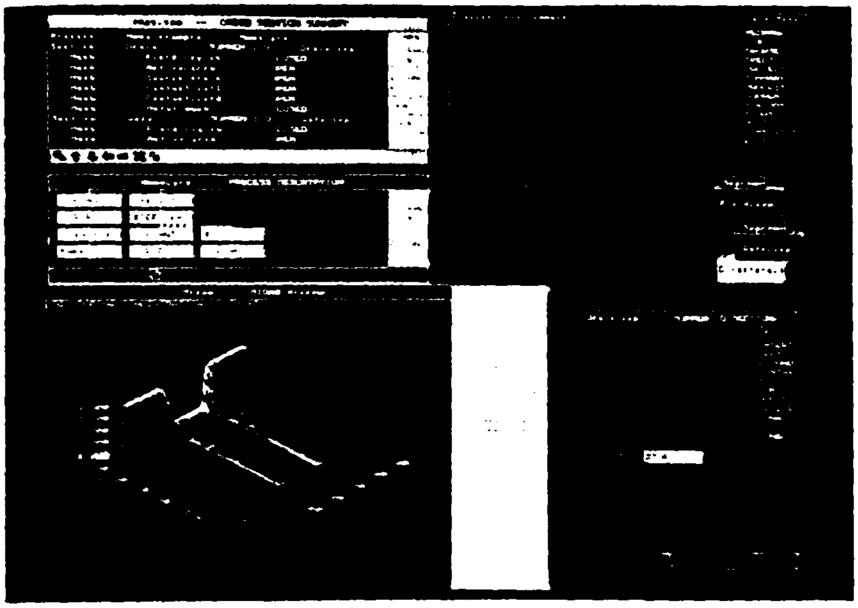
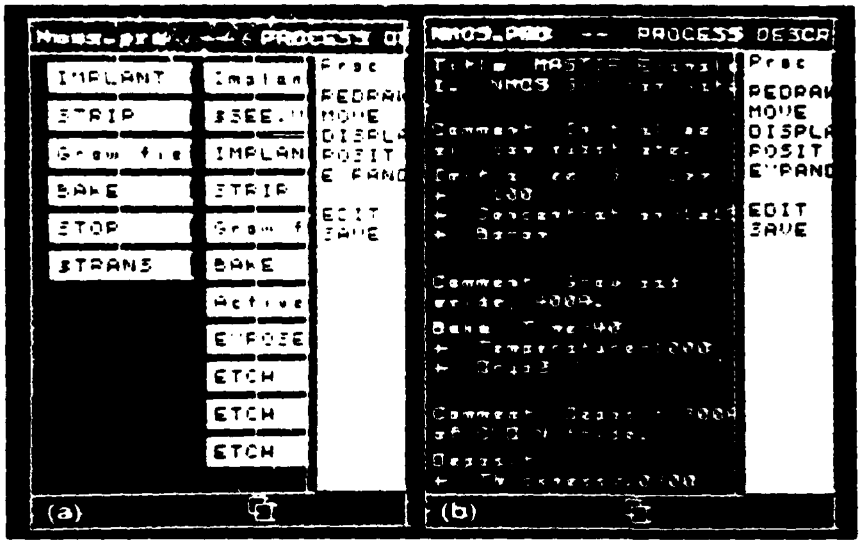


Figure 2. The MASTIF screen as might be seen for an NMOS project. The command area and main menu are in the upper right corner, the cross-section summary window is in the upper left, and the process description window is in the middle left. The Midas window in the lower left shows an electrostatic potential surface plot. The Suprem-III simulation window appears in the lower right.



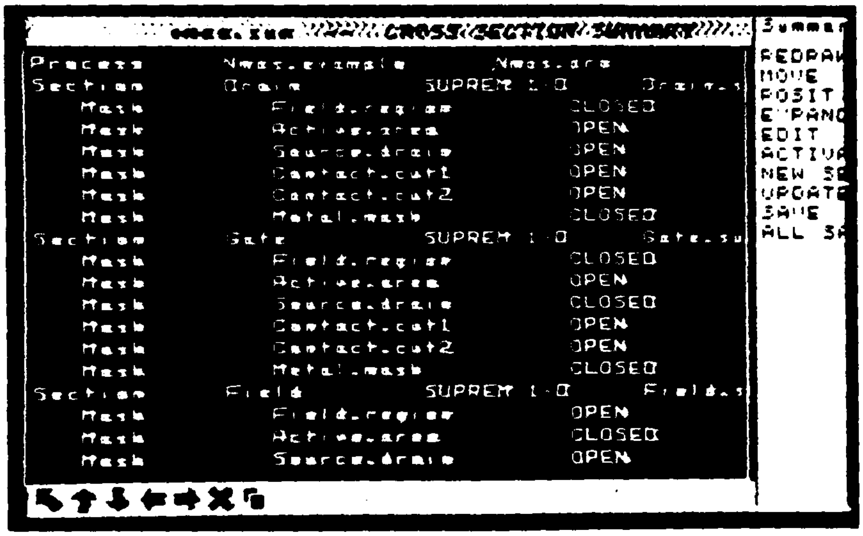
```
Title MASTIF Exampl
I. NMOS [illegible]

Comment [illegible]
[illegible]
[illegible]
+ 100
+ Concentration [illegible]
+ Boron

Comment Grow [illegible]
[illegible] 400A.
Bake Time=40
+ Temperature=1000
+ [illegible]

Comment [illegible] 7000A
[illegible]
Deposit
+ Thickness=0.700
```

Figure 3. MASTIF process description window shown in block display mode (**a**) and full text mode (**b**).



```
Process     Nmos.example          Nmos.arc
Section     Drain          SUPREM 1-D        Drain.s
    Mask        Field.region        CLOSED
    Mask        Active.area         OPEN
    Mask        Source.drain        OPEN
    Mask        Contact.cut1        OPEN
    Mask        Contact.cut2        OPEN
    Mask        Metal.mask          CLOSED
Section     Gate           SUPREM 1-D        Gate.su
    Mask        Field.region        CLOSED
    Mask        Active.area         OPEN
    Mask        Source.drain        CLOSED
    Mask        Contact.cut1        OPEN
    Mask        Contact.cut2        OPEN
    Mask        Metal.mask          CLOSED
Section     Field          SUPREM 1-D        Field.s
    Mask        Field.region        OPEN
    Mask        Active.area         CLOSED
    Mask        Source.drain        OPEN
```

Figure 4. MASTIF cross-section summary window showing the process mask settings for each one-dimensional cross section to be simulated.

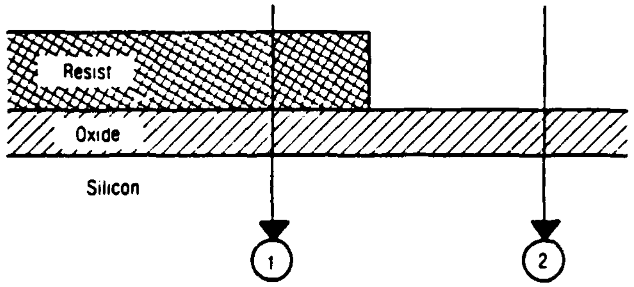


Figure 5. Defining one-dimensional cross sections. In cross-section 1, the resist layer blocks further processing; that is, the mask setting is "closed". In cross section 2, the resist does not block further processing, and the mask setting is then defined as open.

SUPREM-III PLOT

With the plot window, the user can examine the results of a Suprem-III process simulation graphically (only the results of Suprem-III simulations can be plotted with this window). The user can interactively change plot parameters without performing any resimulation.

DEVICE SIMULATION

We have implemented a Minimos simulation window to create and edit input files to the Minimos device simulator. While MASTIF can run Minimos from this window, we have found that the need for an interactive interface to the simulator is not that great, since runs are typically quite time consuming. If we send Minimos runs to a separate computation server, however, such an interface would prove helpful.

MIDAS

The engineer can analyze the results of device simulation through the Midas window, which consists in large part of a Minimos postprocessor. The Midas subsystem is an interactive tool that can present Minimos simulation information in both text and one- and two-dimensional graphics.

MISCELLANEOUS

Other windows include a browse window for perusing text files and a specs window for interactively changing the appearance and configuration of the workstation. Additional main menu functions provide access to the underlying operating system, request directory listings, save the state of MASTIF, and examine the status of the process background simulations. We have also implemented a general-purpose scientific plotting application, called Giraphe, that interacts with the window system.

MASTIF COMPONENTS

Because MASTIF is an integrated workstation, it consists of several hardware and software components. The hardware includes a multiuser VAX 11/750, which runs the VMS operating system, and an AED 767 color graphics display terminal with a digitizing tablet and keyboard.

The software consists of

- individual simulation programs, including Suprem-III and Minimos,

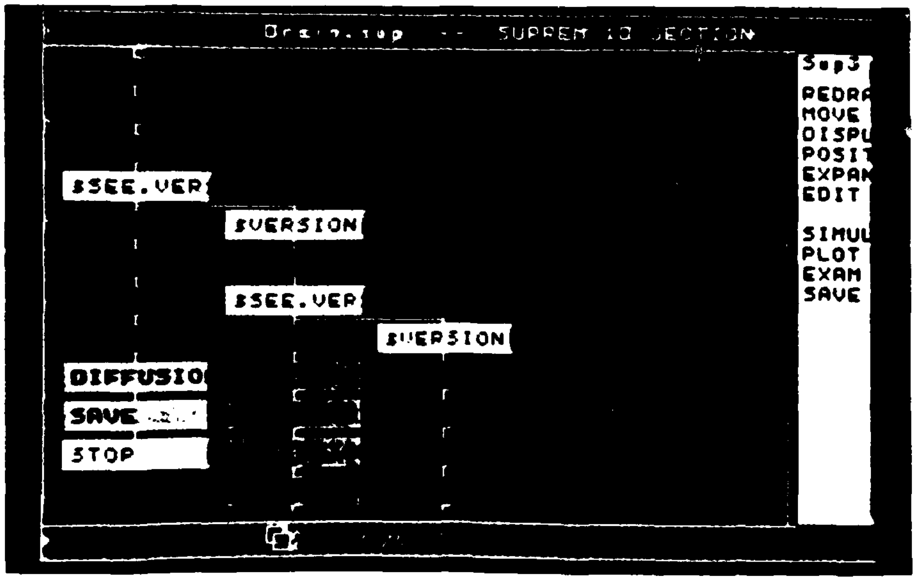


Figure 6. Suprem-III process simulation window. The input file for the gate cross section is in block mode only.

- the MFB graphics package[9] for some degree of device independence
- general MASTIF support modules
- MASTIF application window modules[10]

INFORMATION STORAGE

An important issue in any CAD tool is the choice of data representation. Our overriding concern in the implementation of MASTIF was making the system compatible with existing process and device simulators. As a first step, we had the station serve as a greatly enhanced *simulation environment.* We therefore wanted MASTIF to be able to deal with Suprem-III and Minimos standard input files, as well as with the binary output files of these simulators.

Figure 7 illustrates our plan for information storage in MASTIF, in which information is divided into two types. First, the information from the process description and cross-section summary windows, as well as that from the simulation window are stored as text files. The user reads these files directly, while the Suprem-III simulator reads the files from the Suprem-III simulation window. MASTIF can understand these text files with the help of its syntax and parsing subsystem (described later).

The second type of stored information includes the results of simulations. MASTIF stores the profile structures and device characteristics using the binary format of the simulators. Since the user is likely to interact heavily with MASTIF during process development, the simulated profile structure is saved after each process step. We feel that this need outweighs the penalty of additional file storage.

MASTIF SOFTWARE

MASTIF software uses approximately 25,000 lines of C and Fortran or Ratfor code. Because our goal was to keep MASTIF as portable as possible, we chose conventional languages, such as C and Fortran or Ratfor, and have kept it virtually independent of software packages. Other than the MFB package and each simulation program, the MASTIF code is free of embedded software packages, including the window management subsystem. The modules written to support the MASTIF application windows include the window management subsystem, a syntax and parsing subsystem, and a subprocess handling subsystem.

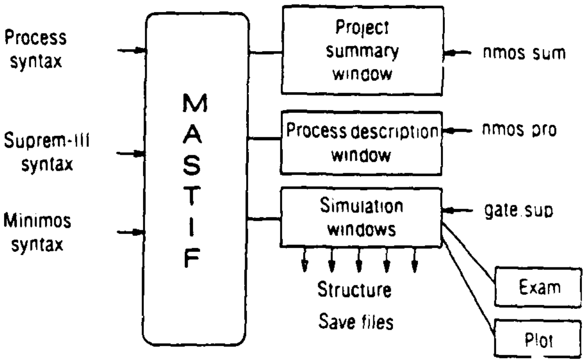


Figure 7. MASTIF information storage. Window information is stored as text files, while simulated profiles are stored in binary format.

WINDOW MANAGEMENT

The window manager, which consists of a display manager, a menu handler, and an I/O manager, performs typical window functions, such as automatic refresh. All the windows are under direct control of MASTIF; they are not autonomous processes. To conserve screen display space, we included a simple window icon facility.

Menus in MASTIF can be either permanent or pop-up and may be textual, iconic, or special-purpose, such as choice according to color. A standard set of icons provides for scrolling and deleting windows and for switching window displays. The user can choose textual menu options either by pointing or by keyboard entry.

MASTIF supports the display and input of text and point information. At a string prompt, it will accept text or point input via the keyboard or a mouse. Users can filter point or string inputs through the command processor before issuing them to calling procedures. With this flexibility, they can execute intermediate commands while in the middle of answering questions to another command. When confronted with a prompt for a file name, the user may request a directory listing before completing a response, for example.

SYNTAX AND PARSING

Most of MASTIF's windows contain information that the user can manipulate. A syntax and parsing subsystem allows the programmer to implement a whole new type of window. The programmer writes the syntax for application input files. Once the syntax is specified, MASTIF reads the text files. After that, the window handlers can display and manipulate them, and they are rewritten as text files for storage.

The format of these input files is similar to that of many process and device simulation programs. It consists of a sequence of statements, each of which begins with a statement name or label. The statement contains named parameters that may be Boolean, character, or numeric. The order of parameters within a statement does not matter, and the case of statements is often unimportant as well. The "+" serves as a continuation character that enables a statement to run onto the next line.

The syntax is a listing of the statements for a particular application; it is a textual specification of the possible statements, as well as the possible parameters for each statement. In addition, the syntax expresses the logical dependencies of these parameters—the programmer can specify mandatory, optional, and mutually exclusive parameters or groups of parameters. Thus, MASTIF provides a simple parameter specification grammar that the programmer may use to express the syntax for some application.

Figure 8a shows an example of the syntax statement for a Suprem-III diffusion line. Parameters enclosed in [ ] are optional, those enclosed in ( ) are mandatory. When a group of parameters is separated by |, the user can specify one and only one of that parameter group in the input file.

Once a syntax file has been written, MASTIF can parse input files for that application. An input line corresponding to the syntax of Figure 8a is shown in Figure 8b. The parameter specification grammar can express the syntaxes of several process and device simulators, and can in general handle input files like those just described. Syntaxes expressed by MASTIF using this grammar include the process description, the Suprem-III input language, the cross-section summary, and the Minimos input language.

```
        Diffusion
          Time=<n> Temperature=<n> [T.Rate=<n:0.0>1]
          [   (Gas.Concentration=<n> | Solidsolubility)
              (Antimony | Arsenic | Boron | Phosphorous)]
          [   (Dry02 | Wet02 | Nitrogen)
              [Pressure=<n>] [P.Rate=<n>] [HCL%=<n>]]
          [Dtmin=<n>] [Dtmax=<n>]
          [Dcmin=<n>] [Dcmax=<n>]
  (a)     [Errmin=<n>] [Errmax=<n>]

        Diffusion    temp=1000 time=30 boron solidsol
  (b)                dtmin=0.01 dtmax=10.0
```

Figure 8. Parameter specification synta:: "›- syntax statement for the Suprem-III diffusion card (**a**) and the input statement corresponding to the syntax statement (**b**).

The input file manipulator provides a menu for creating and editing input files using the syntax structures. The user can check the syntax of lines incrementally by viewing a display in either a full textual mode, or in a simple graphic (block mode) format. We are in the process of constructing a standard form editor for the entry of lines, which shows the possible choices and parameter dependencies.

SUBPROCESS HANDLING

The background/subprocess handling subsystem manages the executions of Suprem-III and Minimos simulations. To maintain the modularity of the system and to make it easy to extend, all simulators are maintained in stand-alone form. MASTIF generates the input files, manages execution, and accesses the results of simulatcrs without requiring any change to the simulation programs themselves. We may in the future be able to execute simulators offline in a networked environment.

RESULTS

We are using MASTIF at MIT to facilitate process and device design and research. We have realized that a highly interactive workstation for process and device design is a valuable tool, but we have also found a number of drawbacks in our current implementation. As a result of working with MASTIF, we have also developed a model of process and device engineering.

AN ENGINEERING MODEL

From the start, we wanted MASTIF to be well-suited to the actual engineering practice of designers. To fulfill that goal, we developed a two-part model of fabrication engineering. In the first part, called incremental process development, the engineer constructs an overall fabrication process primarily a single step at a time. In the second part, called global process development, the engineer modifies the design to satisfy the overall specifications of the process or device and examines long-range goals.

In incremental process development, as the engineer adds a step or short sequence of steps, he wants to evaluate the resulting structure. Process parameters in the most recently added steps need to be modified first to try to satisfy informal intermediate goals. For example, the engineer may desire a very low resistance drain region, so there may be a goal such as "achieve an intermediate sheet resistance of x immediately following implantation and activation." The engineer then chooses implantation parameters through knowledgeable trial and error until this goal is approached. If he limits the range of parameter modification, and allows fairly loose goals, he should have few complications in incremental development.

The engineer then evaluates the results of the incremental phase for the long-range affects of the decisions made. For instance, if the decision was to change implantation energy in an early process step, the engineer needs to see that this may profoundly affect final profile and electrical characteristics.

The global development phase is really just a trade-off analysis. As such, it can be used to evaluate existing processes as well as those developed from scratch during the incremental phase.

Incremental and global process development are similar in that both consist of modify-simulate-analyze loops. The difference is in the size and complexity of those loops. In global development, the range of the loops is larger and the loops are typically harder to manage. The numerous versions that the engineer generates during this phase make it difficult to keep track of what version is doing what. Process parameters must be modified to achieve the performance goals of the device. The modify-simulate-analyze loop then broadens in scope, and becomes a "modify process"- "simulate process"- "simulate device"- "analyze device" loop. Complexity grows as the analysis of the process or the device becomes more thorough, for instance, as we incorporate process sensitivity or yield analyses.

In incremental development, on the other hand, the basic topology of the fabricated structure is the principal goal and is determined largely by the overall sequence of steps. The exact choice of process parameters is less critical at this stage of process development, so the modify-simulate-analyze loop stays fairly small.

A good way to differentiate the two is to think of incremental development as the structural design of the device or profile, while global development is directed at the electrical or device design.

MASTIF aids both these phases of process and device design. In process design, it offers three capabilities. First, it allows the engineer to perform process simulation in single steps or in short sequences of steps in a highly interactive fashion. Second, it allows the engineer to immediately and interactively examine and compare the results of process simulation at any point in the process. And third, it provides version management capabilities in the process description and process simulation windows. With these capabilities, the engineer may add a process step, simulate it, examine the results with a plot window, and try another version of the step—all interactively.

MASTIF offers less complete assistance in device design, primarily because the whole task is unwieldy. MASTIF offers the capabilities just discussed for process design as well as an interactive analysis tool in the Midas window to examine device simulation results. The interfaces between the engineer and the Minimos device simulator as well as between process and device simulators, however, are currently a weak point in the overall system. These interfaces do exist; the engineer can create simulation structures for Minimos, but not interactively.

In implementing MASTIF, we have tried to bring together a set of functions to help process and device design, and we have done that to some extent. Our real contribution, however, is that MASTIF provides a framework for including additional tools as they are developed.

A WORKSTATION APPROACH

Work has been done elsewhere to provide an environment for the integration and use of these tools. In the simplest cases, these environments consist of assorted simulation programs that the user can run and chain together.[11] On the complex end of the spectrum, complete operating systems (or even company-wide networks) are being developed to provide tools and protocols for the use and integration of CAD programs.[12,13]

We have not attempted anything on the scale of these projects. Instead, we chose to construct a single workstation-oriented system that emphasizes user interaction and tight tool integration. We have also built a framework for including more tools as they are developed.

The advantage of this approach is that color, quality graphics, and a consistent window-oriented user interface are possible. The disadvantage is that workstation hardware is expensive. In the typical university or industrial environment, every designer may have a terminal, but not necessarily a color graphics workstation. For us to tie the software to the workstation (as MASTIF does), the engineer must be using the workstation to run individual modules in the MASTIF system. A better approach, perhaps, is to create versions of the individual modules that the engineer can execute from either a simple graphics terminal or a conventional terminal.

There are other problems as well, which stem primarily from MASTIF's window system. The windowing lacks many of the features of full-blown window systems. Moreover, windows provided as part of MASTIF are the only windows available to the user. Other functions normally provided by the operating system are not available. While our implementation of MASTIF provides communicating windows for process and device design, this inability to interact with the underlying operating system is a drawback.

TOOL INTEGRATION

Our primary objective in the MASTIF project was to produce a working tool for process and device design and research. To meet this goal, we tried to integrate existing tools and build new ones. It became painfully obvious we could not get as far as we had planned—and for one big reason: There is no uniform representation for either structure information (wafer profile) or device information. Because we had no general-purpose analysis tools, we could not refine the interfaces between simulation programs themselves. To go any further in providing tools and integrating them, we must have a profile interchange format.

MASTIF has, however, contributed significantly in the understanding of what is needed in a complete design environment. Design and implementation of additional tools to fulfill the needs for specification, synthesis, capture, verification, simulation, and analysis can begin in earnest. Even those limited tools currently incorporated in MASTIF allow the engineer to develop a fabrication process at a higher level of abstraction than before. Continued development of process and device CAD tools will make effective and timely process design a reality.

FUTURE WORK

As we have mentioned, MASTIF has been particularly useful in crystallizing thoughts on the functions needed for process and device design. It has allowed us to identify two important areas that need further work: developing a profile interchange and creating a process description language.

PROFILE INTERCHANGE FORMAT

Work is already in progress to address the need for interchange formats among related CAD tools.[14,15] Work on a common interchange format for process and device design, however, is only in the discussion stage. A powerful, uniform format or representation for profile and device information is critical to the completion of the design environment and would fuel the development of functions that are sorely needed. For example, with a uniform format, the engineer could segment current simulation programs into separate simulation and analysis tools. A verification function for comparison of measured and simulated structures is a second example.

Work is underway to develop a usable interchange format in both ASCII and binary forms. An ad hoc PIF (profile interchange format) standards committee under the direction of Andy Neureuther of UC Berkeley is working to specify a usable format. At MIT, we are investigating a post-processing capability built on this format and are constructing a prototype for general-purpose textual and graphical analysis of information written in that format.[16]

PROCESS DESCRIPTION LANGUAGE

A representation for the fabrication process is a strong requirement for a process and device design workstation. In this research, the representation has been constructed with a view toward describing the process for simulation and analysis. A related issue, however, is the complete capture of a fabrication process, encompassing not only simulation information but also recipe instructions, and all other information needed to define a process completely.[8] We are currently working with the MIT Computer-Aided Fabrication project to develop a process flow language suitable for both fabrication and simulation.

Our work on MASTIF has allowed us to build a foundation in the area of process and device design. We have made two valuable contributions. First we are now certain that a system for process and device design must provide a range of CAD tools. These include not only simulation but also specification, capture, verification, analysis, and synthesis functions. Second, when we implemented a subset of these functions in the MASTIF workstation, the result was more efficient process and device design.

ACKNOWLEDGMENTS

This work has been performed in the MIT Microsystems Technology Laboratory as part of the Computer-Aided Fabrication research project. Support for this research has been provided in part by DARPA under contract N00014-85K-0213 and by the National Science Foundation through a graduate fellowship.

Several people have contributed to the development of MASTIF, both with working code and through valuable discussions. They are Jarvis Jacobs, Thye-Lai Tung, Robert Harris, Rex Lowther, Ron Duncan, and Dave Hamilton.

REFERENCES

1. C.P. Ho et al., "VLSI Process Modeling—SUPREM-III," *IEEE Trans. Electron Devices*, Nov. 1983, pp. 1438-1452.
2. S. Selberherr, A. Schutz, and H.W. Potzl, "MINIMOS—A Two-Dimensional MOS Transistor Analyzer," *IEEE Trans. Electron Devices*, Aug. 1980, pp. 1540-1550.
3. M.E. Daniel and C.W. Gwyn, "CAD Systems for IC Design," *IEEE Trans. Computer-Aided Design*, Jan. 1982, pp. 2-12.
4. K.M. Cham, S. Oh, and J.L. Moll, "Computer-Aided Design in VLSI Device Development," *IEEE J. Solid-State Circuits*, Apr. 1985, pp. 495-500.
5. A.J. Strojwas, "CMU CAM System," *Proc. Design Automation Conf.*, June 1985, pp. 319-325.
6. H.L. Ossher and B.K. Reid, *FABLE: A Programming Language Solution to IC Process Automation Problems*, tech. rpt. 248, Computer Systems Lab, Stanford Univ., Stanford, Calif., 1985.
7. P. Penfield, "Two-Stage Generic Process-Step Model," working paper of CAF project, MIT, Cambridge, Mass., 1985.
8. A.R. Neureuther, C.H. Ting, and C.Y. Liu, "Application of Line-Edge Profile Simulation to Thin-Film Deposition Processes," *IEEE Trans. Electron Devices*, Aug. 1980, pp. 1449-1459.
9. G.C. Billingsley, "Program Reference for KIC," memo UCB/ERL M83/62, ERL, Univ. of Calif.—Berkeley, Berkeley, Calif., Oct. 1983.
10. D.S. Boning, *MASTIF—A Workstation Approach to Integrated Circuit Process and Device Design*, master's thesis, MIT, Cambridge, Mass., May 1986.
11. R.J. Skokel and D.B. MacMillen, "Practical Integration of Process, Device, and Circuit Simulation " *IEEE Trans. Electron Devices*, Oct. 1985, pp. 2110-2116.
12. R.A. Friedenson, J.R. Breiland, and T.J. Thompson, "Designer's Workbench: Delivery of CAD Tools," *Proc. Design Automation Conf.*, pp. 15-22, 1982.
13. A.F. Hutchings, R.J. Bonneau, and W.M. Fisher, "Integrated VLSI CAD Systems at Digital Equipment Corporation," *Proc. Design Automation Conf.*, pp. 543-548, 1985.
14. *EDIF Specification—Version 1 1 0*, Electronic Design Interchange Format Steering Committee, Electronic Industries Assoc., Washington, DC, Nov.1985.
15. M. Sugimoto and M. Fukuma, "Standard Description Form for Device Characteristics in VLSI's," *IEEE Trans. Computer-Aided Design*, Apr. 1986, pp. 293-302.
16. R.B. Duncan, *Implementation of Graphical Analysis and Parsing Utilities for an IC Profile and Device Structure Interchange Format*, MIT, Cambridge, Mass., May 1986.

Duane S. Boning is a National Science Foundation Graduate Fellow at the Massachusetts Institute of Technology. His primary research interests are in CAD for fabrication processes and devices and computer-aided fabrication of ICs. Boning holds a BS in electrical engineering, a BS in computer science, and an MS in electrical engineering and computer science—all from MIT. He is a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi and a student member of the IEEE and ACM.

Dimitri A. Antoniadis is a professor of electrical engineering at MIT and director of the MIT Microsystems Technology Laboratories. His current research includes work on extreme submicron and quantum-effect electronic devices, computer aids for the design and fabrication of silicon devices for ICs, and novel concepts in semiconductor device fabrication. Previously, he was a research associate and instructor at Stanford University.

Antoniadis holds a BS in physics from the National University of Athens and an MS and a PhD in electrical engineering from Stanford University. He has written more than 80 papers on semiconductor devices and IC technology. He is a senior member of the IEEE and a member of the Electrochemical Society and Materials Research Society.

Questions about this article may be directed to D. Boning, MIT, EECS Dept., Cambridge, MA 02139.

Submitted to the 1988 Hilton Head Workshop on Solid-State Transducers

Characterization of the Mechanisms Producing Bending Moments in Polysilicon Micro-cantilever Beams by Interferometric Deflection Measurements

Theresa A. Lober, Jiahua Huang, Martin A. Schmidt, and Stephen D. Senturia
MIT Microsystems Technology Laboratory

The stressed condition of CVD polysilicon films presents design limitations for the development of silicon microstructures, since film warpage determines the maximum free standing lateral dimensions of a suspended structure. Measurements of polysilicon cantilever beam deflection as a function of geometry and process history provide an in situ method for distinguishing the origins of film warpage. We report here the use of a Linnik interferometer to measure vertical deflection of polysilicon cantilevers and bridges (Figs. 1 and 2) with $\equiv$ 50Å resolution. The interference fringes, (Figs. 3 and 4) generated by the passage of coherent light through the interferometer onto suspended structures, follow the profile of deflected structures. The straight fringes in the field regions of Figs. 3 and 4 provide coordinates against which the deflection of the bridges and cantlilevers can be measured (the fringe spacing corresponds to the half-wavelength of the sodium yellow line, 2620 Å). For large deflections, as in Fig. 3, the deflection is measured directly from the photo. For small delfections, as in the nearly flat cantilever in Fig. 4, the angle, θ, between the cantliever and background fringes determines the vertical bridge deflection, δ, as

$$\frac{\delta}{L} = \mathrm{Tan}\theta \frac{\lambda}{2D}$$

where L is the suspension length, λ is the illuminating light's wavelength, and D is the separation between two undeflected fringes. The remaining canitlevers in Fig. 4 have deflected until they are touching the substrate.

The interferometric measurement of cantilever deflection has been employed to assess the effect of doping and annealing on polysilicon microstructure rigidity. A 1μm sacrifical LTO layer was deposited on 4" wafers and patterned. LPCVD poly, 0.5 μm and 1.0 μm was deposited at 625°C. Some samples were $POCl_3$ doped at 925 °C. The samples were then patterned to form cantilevers. Some samples, both doped and undoped, then received an anneal at 1150 °C for 20 minutes. The structures were released using BHF to undercut the oxide spacer layer, and were rinsed in deionized water and methanol.

Two bending moments can induce cantilever deflection, as shown in Fig. 5. The first is due to the clamped pedestal boundary, and causes a deflection linearly dependent on the cantilever length, L. The second is due to stress eccentricity through the beam thickness, and causes a deflection that depends quadratically on L. If the effects were simply additive, the combined effect of both moments would be of the form $\delta \propto K_1L + K_2L^2$. A plot of δ/L vs. L should be a straight line, with an intercept at K_1 reflecting the pedestal moment, and a slope K_2 reflecting the stress eccentricity. The measured deflections of the fabricated cantilever are shown in Fig. 6. Each plot of δ/L vs L reaches a sudden peak at the beam length at which the tip hits the substrate. This large deflection may be caused by effects of surface tension or the bending moments due to the pedestal boundary and doping induced stress eccentricity. The data from shorter cantilevers for undoped samples, both annealed and not annealed, show a constant pedestal moment, and no eccentricity, but the annealed beams show a smaller moment. The doped samples display a slope, characteristic of the bending due to stress eccentricity, but do not fit the simple model above. Either doping, or annealing, or both achieve similiar reduction in stress. Smaller deflections are exhibited by the thicker, doped, annealed film since additional rigidity is gained from added thickness. This study demonstrates that both end-effects and stress eccentricity contribute significantly to the deflections of suspended surface-micromachined polysilicon structures. A simple model explains the observed trend, but is not quantitatively obeyed.